MTI #1789

PROCESSOR WITH DUAL-DEADTIME PULSE WIDTH MODULATION GENERATOR

FIELD OF THE INVENTION:

The present invention relates to a processor having a pulse width modulation (PWM) generator that generates deadtime delays in which the delay from the inactivation of a first signal to the activation of a second signal may be different than the delay from the inactivation of the second signal to the activation of the first signal.

BACKGROUND OF THE INVENTION:

Processors, including microprocessors, digital signal processors and microcontrollers, operate by running software programs that are embodied in one or more series of program instructions stored in a memory. The processors run the software by fetching the program instructions from the series of program instructions, decoding the program instructions and executing them. In addition to program instructions, data is also stored in memory that is accessible by the processor. Generally, the program instructions process data by accessing data in memory, modifying the data and storing the modified data into memory.

Processors may be programmed to perform a wide variety of functions in software. In some cases, however, dedicated hardware may be included in a processor that significantly eases the processing load needed to perform certain functions. This allows the use of lower performance processor for these functions, which lowers the cost of the processor. One type of dedicated hardware that may advantageously be included in a processor is power control hardware. Power control hardware provides the capability to control circuitry and devices that use significant amounts of power. For example, power control hardware may be used to control motors, power supplies, etc.

One common mode of operation of power control hardware is pulse width modulation (PWM). In PWM, the power level is controlled by controlling the duty cycle of a signal that has only two states - active and inactive. The signal is then integrated in a device, such as a motor or a capacitor, to yield the equivalent of a continuously varying voltage and current.

When PWM hardware is included in a processor, external switching devices, such as transistors, must be used in order to handle significant amounts of power. These switching devices have less than perfect switching characteristics, especially when connected to devices such as motors. Problems arise with conventional PWM hardware, which has been included in current processors, in dealing with the less than perfect switching characteristics of connected switching devices.

A need arises for a technique that provides an improved ability to deal with the less than perfect switching characteristics of external switching devices that are connected to PWM hardware included in a processor.

SUMMARY OF THE INVENTION:

The present invention is a processor that has pulse width modulation generation circuitry that provides an improved ability to deal with the less than perfect switching characteristics of external switching devices that are connected to PWM hardware included in a processor. This is accomplished by providing complementary PWM output signals that have a deadtime delay, which is a delay between the inactivation of one signal and the activation of the complementary signal. The deadtime provides the capability to deal with the switching characteristics of external switching devices. Advantageously, the present invention provides dual deadtime capability. That is the delay between the inactivation of the first signal and the activation of the second signal may be different than the delay between the inactivation of the second signal and the activation of the first signal. This provides an improved capability to deal with non-symmetric switching characteristics of the external switching devices, and the circuitry to which they are connected.

According to one embodiment of the present invention, a dual deadtime pulse width modulation generator for a processor includes deadtime generation circuitry operable to generate a first pulse width modulated signal and a second pulse width modulated signal complementary to the first pulse width modulated signal, wherein there is a first delay between inactivation of the first pulse width modulated signal and activation of the second pulse width modulated signal, a second delay between inactivation of the second pulse width modulated signal and activation of the first pulse width modulated signal, and the first and second delays are not equal.

In one aspect of the present invention, the first delay and the second delay may be independently settable.

According to one embodiment of the present invention, a dual deadtime pulse width modulation generator for a processor includes pulse width modulation generation circuitry operable to generate a first pulse width modulated signal, and deadtime generation circuitry including first circuitry operable to generate a second pulse width modulated signal from the first pulse width modulated signal and a third pulse width modulated signal from the first pulse width modulated signal, the third pulse width modulated signal complementary to the second pulse width modulated signal, and second circuitry operable to generate a first delay between inactivation of the second pulse width modulated signal and activation of the third pulse width modulated signal and a second delay between inactivation of the third pulse width modulated signal and activation of the second pulse width modulated signal, wherein the first and second delays are not equal.

In one aspect of the present invention, the second circuitry may include a first register operable to store a value for setting the first delay and a second register operable to store a value for setting the second delay. The second circuitry may further include a first edge detector operable to detect a first edge of the first pulse width modulated signal and initiate generation of the first delay and a second edge detector operable to detect a second edge of the first pulse width modulated signal and initiate generation of the second delay.

In one aspect of the present invention, the deadtime generation circuitry may further include third circuitry operable to generate a first clock signal that is input to the second circuitry for generating the first and second delays. The third circuitry may include a prescaler operable to generate the first clock signal by prescaling a second clock signal. The third circuitry may further include circuitry operable to set the prescaler to a first prescale setting to generate the first clock signal for generating the first delay and to set the prescaler to a second prescale setting to generate the first clock signal for generating the second delay.

According to one embodiment of the present invention, a processor includes deadtime generation circuitry operable to generate a first pulse width modulated signal and a second pulse width modulated signal complementary to the first pulse width modulated signal, wherein there is a first delay between inactivation of the first pulse width modulated signal and activation of the second pulse width modulated signal, a second delay between inactivation of the second pulse width modulated signal and activation of the first pulse width modulated signal, and the first and second delays are not equal.

In one aspect of the present invention, the first delay and the second delay may be independently settable.

According to one embodiment of the present invention, a processor includes pulse width modulation generation circuitry operable to generate a first pulse width modulated signal, and deadtime generation circuitry including first circuitry operable to generate a second pulse width modulated signal from the first pulse width modulated signal and a third pulse width modulated signal from the first pulse width modulated signal, the third pulse width modulated signal complementary to the second pulse width modulated signal, and second circuitry operable to generate a first delay between inactivation of the second pulse width modulated signal and activation of the third pulse width modulated signal and a second delay between inactivation of the third pulse width modulated signal and activation of the second pulse width modulated signal, wherein the first and second delays are not equal.

In one aspect of the present invention, the second circuitry may include a first register operable to store a value for setting the first delay and a second register operable to store a value for setting the second delay. The second circuitry may further include a first edge detector operable to detect a first edge of the first pulse width modulated signal and initiate generation of the first delay and a second edge detector operable to detect a second edge of the first pulse width modulated signal and initiate generation of the second delay.

In one aspect of the present invention, the deadtime generation circuitry may further include third circuitry operable to generate a first clock signal that is input to the second circuitry for generating the first and second delays. The third circuitry may include a prescaler operable to generate the first clock signal by prescaling a second clock signal. The third circuitry may further include circuitry operable to set the prescaler to a first prescale setting to generate the first clock signal for generating the first delay and to set the prescaler to a second prescale setting to generate the first clock signal for generating the second delay.

BRIEF DESCRIPTION OF THE FIGURES:

Fig. 1 depicts a functional block diagram of an embodiment of a processor chip within which the present invention may find application.

Fig. 2 depicts a functional block diagram of a pulse width modulation (PWM) module for use in a processor, such as that shown in Fig. 1.

Fig. 3 depicts an exemplary external circuit, which may be driven by complementary PWM signals generated by the circuit shown in Fig. 2.

Fig. 4 illustrates the concept of deadtime in complementary PWM signals.

Fig. 5 depicts an exemplary block diagram of a deadtime generator circuit shown in Fig 2.

Fig. 6 illustrates the function of the deadtime generator circuit shown in Fig. 5.

DETAILED DESCRIPTION:

Fig. 1 depicts a functional block diagram of an embodiment of a processor chip within which the present invention may find application. Referring to Fig. 1, a processor 100 is coupled to external devices/systems 140. The processor 100 may be any type of processor including, for example, a digital signal processor (DSP), a microprocessor, a microcontroller, or combinations thereof. The external devices 140 may be any type of systems or devices including input/output devices such as keyboards, displays, speakers, microphones, memory, or other systems which may or may not include processors. Moreover, the processor 100 and the external devices 140 may together comprise a stand alone system.

The processor 100 includes a program memory 105, an instruction fetch/decode unit 110, instruction execution units 115, data memory and registers 120, peripherals 125, data I/O 130, and a program counter and loop control unit 135. The bus 150, which may include one or more common buses, communicates data between the units as shown.

The program memory 105 stores software embodied in program instructions for execution by the processor 100. The program memory 105 may comprise any type of nonvolatile memory such as a read only memory (ROM), a programmable read only memory (PROM), an electrically programmable or an electrically programmable and erasable read only memory (EPROM or EEPROM) or flash memory. In addition, the program memory 105 may be supplemented with external nonvolatile memory 145 as shown to increase the complexity of software available to the processor 100. Alternatively, the program memory may be volatile memory, which receives program instructions from, for example, an external non-volatile memory 145. When the program memory 105 is nonvolatile memory, the program memory may be programmed at the time of manufacturing the processor 100 or prior to or during implementation of the processor 100 within a system. In the latter scenario, the processor 100 may be programmed through a process called in circuit serial programming.

The instruction fetch/decode unit 110 is coupled to the program memory 105, the instruction execution units 115, and the data memory 120. Coupled to the program memory 105 and the bus 150 is the program counter and loop control unit 135. The instruction fetch/decode unit 110 fetches the instructions from the program memory 105 specified by the address value contained in the program counter 135. The instruction fetch/decode unit 110 then decodes the fetched instructions and sends the decoded instructions to the appropriate execution unit 115. The instruction fetch/decode unit 110 may also send operand information including addresses of data to the data memory 120 and to functional elements that access the registers.

The program counter and loop control unit 135 includes a program counter register (not shown) which stores an address of the next instruction to be fetched. During normal instruction processing, the program counter register may be incremented to cause sequential instructions to be fetched. Alternatively, the program counter value may be altered by loading a new value into it via the bus 150. The new value may be derived based on decoding and executing a flow control instruction such as, for example, a branch instruction. In addition, the loop control portion of the program counter and loop control unit 135 may be used to provide repeat instruction processing and repeat loop control as further described below.

The instruction execution units 115 receive the decoded instructions from the instruction fetch/decode unit 110 and thereafter execute the decoded instructions. As part of this process, the execution units may retrieve one or two operands via the bus 150 and store the result into a register or memory location within the data memory 120. The execution units may include an arithmetic logic unit (ALU) such as those typically found in a microcontroller. The execution units may also include a digital signal processing engine, a floating point processor, an integer processor, or any other convenient execution unit.

The data memory and registers 120 are volatile memory and are used to store data used and generated by the execution units. The data memory 120 and program memory 105 are preferably separate memories for storing data and program instructions respectively. This format is a known generally as a Harvard architecture. It is noted, however, that according to the present invention, the architecture may be a Von-Neuman architecture or a modified Harvard architecture, which permits the use of some program space for data space. A dotted line is shown, for example, connecting the program memory 105 to the bus 150. This path may include logic for aligning data reads from program space such as, for example, during table reads from program space to data memory 120.

A plurality of peripherals 125 on the processor may be coupled to the bus 125. The peripherals may include pulse width modulation (PWM) module 160 and other peripherals 165, such as analog to digital converters, timers, bus interfaces and protocols such as, for example, the controller area network (CAN) protocol or the Universal Serial Bus (USB) protocol and other peripherals. The peripherals exchange data over the bus 150 with the other units. The PWM module 160 is capable of generating multiple, synchronized pulse width modulated (PWM) outputs. The PWM module 160 may be advantageously applied to a variety of power and motion control applications, such as control of Three-Phase AC Induction Motors, Switched Reluctance (SR) Motors, Brushless DC (BLDC) Motors, and Uninterruptable Power Supplies (UPSs).

The data I/O unit 130 may include transceivers and other logic for interfacing with the external devices/systems 140. The data I/O unit 130 may further include functionality to permit in circuit serial programming of the Program memory through the data I/O unit 130.

Fig. 2 depicts a functional block diagram of a pulse width modulation (PWM) module 160, for use in a processor 100, such as that shown in Fig. 1. PWM module 160 includes control registers 202, timebase 204, special event logic 206, at least one pulse width modulation generator, such as PWM generators 208A, 208B, 208C, and 208D, for each PWM generator, a deadtime generator, such as deadtime generators 210A, 210B, 210C, and 210D, and output driver circuitry 212. Bus 150 communicates data among units of processor 100 and elements of PWM module 160. In particular, bus 150 communicates data with control registers 202, timebase 204, special event logic 206, and the at least one PWM generator, such as PWM generators 208A, 208B, 208C, and 208D.

Control registers 202 store values that are modifiable in software and provide the capability to control and configure the operation of the elements of PWM module 160. Control registers 202 may include a plurality of control registers, each control register including a plurality of bits. Each control register may be read, under software control, to determine the configuration and operational state of elements of PWM module 160.

Likewise, each control register may be written, under software control, to set the configuration and control the operation of element of PWM module 160. For example, bits in control registers 202 may provide the capability to enable or disable the generation of PWM signals by PWM module 160. Bits in control register 202 may provide the capability to set the polarity and timing of signals output by PWM module 160, such as the frequency, duty cycle, and dead time of such signals. Bits in control registers 202 may provide the capability to enable, disable, and configure special event triggering, fault triggering, override operation, and other functions of PWM module 160.

Timebase 204 generates timing signals that are used by other elements of PWM module 160, such as special event logic 206 and the PWM generators 206A-D. Timebase 204 may include registers, counters, comparators, and other circuitry that operate with a timing clock signal to provide the capability to generate timing signals having programmable parameters. For example, timebase 204 may provide the capability to program parameters such as: the count direction of included counters, the resolution and prescaling of the timing clock used to generate the timebase signals, the mode of operation of timebase interrupts, postscaling of timebase signals, and the mode of operations of the timebase, such as continuous, free running, single shot, etc.

Special event logic 206 generates trigger signals that provide the capability to synchronize operations external to PWM module 160 with the operation of PWM module 160. For example, in an embodiment of processor 100 in which other peripherals 165 includes one or more analog to digital (A/D) converters, the operation of such A/D converters may be synchronized to the operation of PWM module 160 using the trigger signals generated by special event logic 206. Special event logic 206 uses signals generated by timebase 204 to generate trigger signals that are synchronized with selected points in the period of the PWM signals generated by PWM module 160.

Each PWM generator generates a PWM signal, which is input to a deadtime generator. Each PWM generator, such as PWM generator 208A, may include a duty cycle register, such as duty cycle register 214, a comparator, such as comparator 216, and associated circuitry. Duty cycle register 214 stores a value that controls the duty cycle of the PWM signals. The duty cycle of a PWM signal is the fraction of each complete PWM cycle that the signal is in the active state. Duty cycle register 214 typically includes a buffer register, which is accessible by software, and a comparison register, which stores the actual compare value used in each PWM cycle. The value in the comparison register is compared by comparator 216, to a value generated by timebase 204. The status of this comparison controls the signals output from comparator 216, which, in turn, control whether the PWM signal is in the active or inactive state.

The output from each comparator, such as comparator 216, is input to a deadtime generator, such as deadtime generator 210A. Deadtime generator 210A may pass the signal from comparator 216 without alteration, or deadtime generator 210A may alter the signal. Deadtime generator 210A may generate a set of complementary PWM signals based on the signal from comparator 216. Complementary signals are signals that are arranged so that when one signal is active, the other signal is inactive. When the active signal becomes inactive, the inactive signal becomes active, and so on. Deadtime generator 210 also inserts deadtime into the complementary signals. Deadtime is a period during which neither complementary signal is active.

The complementary PWM signals generated by each deadtime generator, such as deadtime generator 210A, is input to output driver circuitry 212, which includes circuitry of sufficient capacity to drive circuitry external to processor 100. The drive signals are supplied to external circuitry via processor pins, such as pins 218A and 218B.

Fault inputs 220A and 220B provide the capability to drive one or more of the PWM outputs to a defined state. Such a function is useful if a fault occurs in the external circuitry that is controlled by the PWM outputs of processor 100. The function of the fault inputs is performed directly in hardware, so that fault events can be managed quickly. Examples of faults that may occur include failure of an external switching device, such as a transistor, short circuit of external circuitry or devices, such as a motor, overcurrent detected in external circuitry or devices, a fault in the power supply, etc. Fault pin priority logic 222 provides the capability to prioritize the function of the fault inputs if more than one input becomes active. The signals output from fault pin priority logic 222 are input to the deadtime generators 210A - 210D. The deadtime generators also include fault override logic that overrides the function of the deadtime generator in response to a fault signal from fault pin priority logic 222, if so configured.

Included in control registers 202 are registers that control the configuration and function of PWM module 160 in response to activation of one or more fault inputs. In particular, the registers provide the capability to define whether a particular pair of PWM outputs associated with a deadtime generator, such as PWM outputs 218A and 218B and deadtime generator 210A, are controlled by the fault inputs. If enabled, the override logic in the deadtime generator will respond to a fault output signal 224 from fault pin priority logic 222 and perform a defined action.

Control registers 202 store values that define the state of each PWM output in response to a fault signal input to each fault input. Each PWM output can be defined to be driven inactive or active in response to the fault signal input to each fault input. For example, PWM output 218A may be defined to be driven inactive in response to a fault signal on fault input 220A and may be defined to be driven active in response to a fault signal on fault input 220B. If a PWM output pair associated with one deadtime generator is in the complementary output mode and both PWM outputs are defined to be driven to the active state in response to a fault signal input to a fault input, both PWM outputs would be driven active, which is not desirable. In this situation, the override logic in the deadtime generator will give priority to one PWM output, drive that PWM output active, and drive the other PWM output inactive.

Fault pin priority logic 222 provides prioritization among the fault inputs. If more than one fault input has been defined to control a particular PWM output and at least two such fault inputs become active concurrently, fault pin priority logic 222 selects one of the fault inputs to be given priority. The PWM outputs are driven to the fault states defined for the fault input that has been given priority, and the other fault inputs are ignored. Fault priority logic 222 generates a fault output signal 224 that indicates the selected fault input. Fault output signal 224 in input to the deadtime generators, such as deadtime generator 210A, which drives its associated PWM outputs to the fault state defined for the selected fault input.

Each of the fault inputs has two modes of operation:

Latched Mode: When the fault input is driven active, the PWM outputs will remain in the defined fault states until the fault input is driven inactive and the fault condition is cleared in software. The PWM outputs will be enabled for normal, non-fault operation once the fault condition is cleared in software.

Cycle-by-Cycle Mode: When the fault input is driven active, the PWM outputs will remain in the defined fault states until the fault input is driven inactive. When the fault input is driven inactive, the PWM outputs will return to normal, no-fault operation at the beginning of the next PWM period.

The mode of operation of each fault input is defined in registers included in control registers 202.

Each fault input may also be controlled directly by software. Processor 100 can be configured so that software can directly drive the active or inactive levels of each fault input.

An example of an external circuit that may be driven by complementary PWM signals is shown in Fig. 3. In this example, three transistor pairs, a first pair including transistors 302A and 302B, a second pair including transistors 304A and 304B, and a third pair including transistors 306A and 306B, are connected to complementary PWM outputs of processor 100, either directly or via appropriate additional circuitry. For example, the signal on pin 218A, shown in Fig. 2, may be connected to input 308A of transistor 302A and the complementary signal on pin 218B, also shown in Fig. 2, may be connected to input 308B of transistor 302B. One of skill in the art would recognize that pins 218A and 218B would typically be connected to inputs 308A and 308B, respectively, via appropriate, and well-known, circuitry. Other complementary PWM outputs from processor 100 may similarly be connected to inputs to other transistor pairs.

The output from each transistor pair is formed at a connection between the transistors in the pair. In the example shown, which uses transistors that are MOSFETs, the output of each transistor pair is formed at the connection between the source of the upper transistor and the drain of the lower transistor. For example, output 310A is formed at the connection of the source of transistor 302A and the drain of transistor 302B. The outputs of the transistor pairs are connected to windings of motor 312 and supply of power that drives motor 312.

In the example of Fig. 3, MOSFET transistors and a three-phase AC induction motor are illustrated. One of skill in the art would, of course, recognize that other types of transistors and other types of motors could be used as well as those illustrated. For example, transistors, such as bipolar transistors, insulated-gate bipolar transistors, and other well-known types of transistors, or motors, such as switched reluctance (SR) motors, or brushless DC (BLDC) motors could be used instead of those illustrated, with well-known modifications to the circuitry. Likewise, as is well-known, the PWM signals could be used to control other applications, such as switching power supplies, etc.

The concept of deadtime in complementary PWM signals is illustrated in Fig. 4. Complementary PWM signals are applied to the inputs of a transistor pair. In particular, PWM signal 402A is applied to input 404A of transistor 406A and PWM signal 402B, which is complementary to signal 402A, is applied to input 404B of transistor 406B. Signal 402A has an active period 408, during which signal 402B, being complementary, is inactive. During period 408, transistor 406A conducts and transistor 406B does not conduct, causing the output 410 from the transistor pair to output signal 412. At the end of period 408, signal 402A becomes inactive, then, in period 414, signal 402B becomes active. During period 414, transistor 406B conducts and transistor 406A does not conduct, causing the output 410 from the transistor pair to output signal 416. As is well-known, the rise and fall times of signals 402A and 402B are not infinitely fast, nor are the switching times of transistors 406A and 406B. If signal 402B were to become active immediately upon signal 402A becoming inactive, the finite rise and fall times of the signals and the finite switching times of the transistors would cause a period during which both transistor 406A and transistor 406B were conducting. During this period, current would be conducted directly from power to ground, limited only by the conduction resistances of the transistors. At a minimum, this would waste power and cause unnecessary power dissipation in the transistors. In many cases, it is possible that catastrophic failure of one or more transistors may occur.

This situation can be averted by providing a deadtime, such as deadtime 418, which is a period during which neither complementary signal is active. Thus, when signal 402A becomes inactive, a time delay occurs before signal 402B becomes active. This time delay may be determined based on the rise and fall times of signals 402A and 402B, the switching times of transistors 406A and 406B, and based on parameters of other circuit components, such as motor winding inductances, etc. The provision of a deadtime prevents the power waste, unnecessary power dissipation, and possible catastrophic failure that would otherwise occur.

At the end of period 414, signal 402B becomes inactive, then, in period 418, signal 402A becomes active. During period 418, transistor 406A conducts and transistor 406B does not conduct, causing the output 410 from the transistor pair to output signal 420. Again, a deadtime 422 is provided between the inactivation of signal 402B and the activation of signal 402A. In some embodiments, deadtime 418 may be equal in time to deadtime 422. However, in other embodiments, it is necessary that deadtime 418 have a different duration from deadtime 422. Thus, it is advantageous to provide the capability to configure deadtime 418 independently of deadtime 422.

An exemplary deadtime generator circuit 500, which is suitable for use in pulse width modulation module 106, shown in Fig. 2, is illustrated in Fig. 5. Deadtime generator 500 provide the capability to generate complementary PWM signals that have deadtimes that are independently configurable. For illustration in Fig. 5, the two deadtimes are termed deadtime A and deadtime B. Deadtime generator 500 includes two deadtime registers, deadtime A register 502A and deadtime B register 502B, counter 504, comparator 506, two edge detectors 508A and 508B, and clock control and prescaler circuitry 510. The duration of each deadtime period is selected in software by selecting an appropriate prescaler setting and an appropriate deadtime register setting. Clock control and prescaler circuitry 510 generates a prescaled clock 512 that is output to counter 504. This allows a plurality of ranges of deadtime durations to be selected. Clock control and prescaler circuitry 510 has inputs that include clock 514, deadtime select 516, and prescale selects, prescale select A 518A and prescale select B 518B. Clock 514 is typically derived from a processor or system clock and is input to the prescaler. Prescale select A 518A and prescale select B 518B provide the capability to select different prescaler setting for each deadtime. Deadtime select 516 provides the capability to select desired configurations of clock control and prescaler circuitry 510.

Deadtime A register 502A and deadtime B register 502B are settable by software and store values that control the duration of each deadtime period. Counter 504 counts cycles of the prescaled clock. 512 that is output from clock control and prescaler circuitry 510. For each deadtime period, counter 504 is loaded with the number of cycles that it is to count from either deadtime A register 502A or deadtime B register 502B. The deadtime register that is to be used for each particular load of counter 504. as well as the prescaler setting. The PWM signal 520 that is output from comparator 216 of Fig. 2 is input to edge detectors 508A and 508B. If the PWM edge event is a rising edge, the edge is detected by rising edge detector 508A, which activates circuitry 522A that loads the value stored in deadtime A register 502A into counter 504. Detection of the rising edge of signal 520 also causes prescale A setting 518A to be used by the prescaler 510. If the PWM edge event is a falling edge, the edge is detected by falling edge detector 508B, which activates circuitry 522B that loads the value stored in deadtime B register 502B into counter 504. Detection of the falling edge of signal 520 also causes prescale B setting 518B to be used by the prescaler 510.

Once a value has been loaded into counter 504 and prescaler 510 has been set, counter 504 counts cycles of prescaled clock 512 until a terminal value is reached. The terminal value is detected by comparator 506. Until the terminal value is reached, comparator 506 outputs a signal to circuitry 524, which deactivates the PWM output. When the terminal value is reached, comparator 506 outputs a signal to circuitry 524 that activates the PWM output. Note that circuitry 524 generates complementary PWM output signals from PWM signal 520.

The function of deadtime generator circuit 500 is illustrated in Fig. 6. The signals shown include PWM signal 520, PWM output A signal 602A, and PWM output B signal 602B. A rising edge in PWM signal 520 causes deadtime A 604A to be inserted in PWM output A signal 602A before signal 602A becomes active. No delay is inserted before PWM output B signal 602B becomes inactive. A falling edge in PWM signal 520 causes deadtime B 604B to be inserted in PWM output B signal 602B before signal 602B becomes active. No delay is inserted before PWM output A signal 602A becomes inactive.

While specific embodiments of the present invention have been illustrated and described, it will be understood by those having ordinary skill in the art that changes may be made to those embodiments without departing from the spirit and scope of the invention.